

*REMARKS*

The Office Action of July 28, 2004 has been carefully considered along with the cited references. In view of the foregoing amendments and the following remarks, it is believed that the application should be allowable.

The Office Action asserted that the title of the invention is not descriptive, and that a new title is required. The Office Action did not provide an explanation of why the original title was considered non-descriptive, and applicants maintain that the original title is both concise and clearly indicative of the claimed invention. Nevertheless, in the spirit of cooperation, applicants have changed the title to:

**SIMPLIFIED DEVICE DRIVERS FOR HARDWARE DEVICES OF A  
COMPUTER SYSTEM**

It is believed that this new title is clearly indicative of the invention to which the claims are directed. If, however, the Examiner should again raise the same objection regarding the title, the Examiner is respectfully requested to explain clearly why the title is considered not descriptive, and, if possible, provide suggestion for a better title.

The Office Action also referred to a "preferred format" for claims that numbers each line of every claim, with each claim beginning with line 1, and requires that all future correspondence include the recommended line numbering. Although applicants could not find any reference to such preferred claim format in the MPEP, in the spirit of cooperation, applicants have applied such line numbering in this Amendment. The Examiner is respectfully requested, however, to identify the source of this requirement for future references.

The Office Action objected to the specification because the reference numerals "70" and "80" have been used inconsistently in page 13 of the specification. Applicants have amended the specification to correct this informality. The Office Action also pointed out that the reference numeral "70" in FIG. 2 was not mentioned in the specification. In response, applicants have amended the specification in page 9 such that "70" refers to the "system architecture" of the invention as illustrated in FIG. 2.

The Office Action also objected to the drawings because the reference numeral "70" was not mentioned in the specification, and indicated that a drawing correction would be required. Nevertheless, the amendment of the specification as described links the reference numeral to the system architecture. As a result, the objection to the drawings is rendered moot, and there is no need to amend the drawings.

Turning now to the claims, the Office Action rejected claims 1-2, 6-10, 12-13, 16-17 and 21 under 35 U.S.C. § 103(a) as being unpatentable over Dinallo (U.S. Patent 5,727,212). The Office Action further rejected claims 3-5, 11, 14-15, and 18-20 under 35 U.S.C. § 103(a) as being unpatentable over Dinallo in view of Applicant Admitted Prior Art. As to independent claims 1, 8, 12, and 16, the Office Action asserted that Dinallo teaches all the claim limitations, including first and second device drivers and a device driver interface, except that the device driver is for controlling a hardware device. Specifically, the Office Action called an OS/2 DDTransport shown in FIG. 7 as the first device driver, an OS/2 DDTransport shown in FIG. 2 as the second device driver, and an OO interface (col. 4, lines 31-32) as the device driver interface.

To facilitate an understanding of the invention and how it differs from the cited art, a short summary of the claimed invention is provided here. The present invention is directed to a way to simplify the efforts for hardware vendors to develop reliable device drivers for hardware devices for a computer system. In accordance with the invention, a system framework is provided that enables the use of simplified device drivers that are much easier for the hardware vendors to develop than regular device drivers. A simplified device driver only has to implement entry point functions for a small set of pre-selected operation commands for operations "generic" to different device models and brands of that given device type. For instance, in an example given in the specification, the device type may be a flatbed scanner, and the simplified device driver only has to implement entry point functions for operation commands generic to flatbed scanners, such as "MicroEntry," "GetScanInfo," "SetPixelWindow," and "scan." See, Specification, pp. 16-17. A simplified device driver for a hardware device of a given type, such as a flatbed scanner, works with a common driver provided for that given type, and together they function like a regular device driver. When an application makes a request for an operation by the device, the request is passed through a device driver interface (DDI) to the common driver. The common driver then calls the entry point functions in the simplified device driver to control the hardware device to carry out the requested operation. Because a simplified device driver only has to implement a small number of entry point functions for generic device operation commands, it is significantly less complicated than a regular device driver that has to handle various driver interface functions required by the operating system. As a result, it is much easier for a hardware vendor to develop a high-quality simplified device driver.

Applicants submit that this concept of simplified device drivers implementing only entry functions for a small set of operation commands generic to hardware devices of a given hardware device type is neither taught nor suggested by Dinallo. Dinallo is directed to the problem of integrating object-oriented programming (OOP) software components with procedural programming software components. Col. 1, lines 50-57. The solution provided by Dinallo is to use an OO Generic Device interface 54 to interface the object oriented subsystem/component 50 with procedural device drivers 52. FIG. 2; col. 3, line 66 – col. 4, line 10. The device interface 54 uses OOP methods to wrap around the procedural code from the device drivers, thus allowing communications to occur between the OO component 50 and the procedural device driver 52. Col. 4, lines 20-23. The device interface 54 comprises an OO class named DDInterface and another OO class named DDTransport. Col. 4, line 23-25. It can be seen that Dinallo is not analogous to the claimed invention, due to its focus of interfacing OOP and procedural components. As a result, Dinallo does not teach or suggests the concept of simplified device drivers for hardware devices.

More particularly, applicants submit that the rejection of the independent claims 1, 8, 12, and 16 under Section 103 is not supported by the teachings of Dinallo. For ease of reference, claim 1, which is representative of the independent claims, is reproduced here:

1. A computer-readable medium having computer-executable components for controlling a hardware device of a given device type installed in a computer system, comprising:

a **first device driver** for interacting with, through a device driver interface, an application running on the computer system; and

a **second device driver** programmed to support **entry point functions corresponding to a pre-selected set of operation commands generic to hardware devices of the given device type**, the entry point functions callable by the first device driver for controlling operations of said hardware device,

the first device driver programmed for receiving, through the device driver interface, a request from the application for a requested operation by the hardware device, and calling the entry point functions of the second device driver to control the hardware device to perform the requested operation.

First, applicants points out that the Office Action's labeling the OS/2 DDTransport of FIG. 7 of Dinallo as the first device driver and the DDTransport of FIG. 2 of Dinallo as the second device driver is improper. Dinallo does not describe the OS/2 DDTransport and the DDTransport as two separate device drivers that interact with each other. In fact, the OS/2 DD Transport is the DD Transport in a particular embodiment in which the operating system is OS/2. Moreover, the OS/2 DDTransport and DDTransport are not even device drivers according to Dinallo. As clearly described in Dinallo, DDTransport and DDInterface are two OOP classes of the OO generic device interface 54 that bridges OOP components with procedural device drivers. Since the Office Action already identified the OO interface 54 as the "device driver interface" of the claims, it would not be correct to label the DDTransport class of the OO interface 54 as either of the first and second device drivers.

Moreover, the Office Action overlooked an important limitation of the claims. As called for in the claims, the second device driver is programmed to support "entry point functions corresponding to a pre-selected set of operation commands generic to hardware devices of the given device type" that are callable by the first device driver for controlling operations of said hardware device. This limitation is not taught or suggested by Dinallo. As mentioned above, Dinallo is concerned with interfacing OOP components with procedural device drivers, and does not have any teaching relevant to the concept of using simplified device drivers that are required to support only the entry point functions of a pre-selected set of operation commands generic to

hardware devices of a given type. In this regard, the Office Action cited FIG. 3; col.2, lines 9-13; col. 3, lines 22-23; and col. 4, lines 28-38 as purportedly teaching this limitation. Applicants have carefully reviewed those cited portions of Dinallo, and do not find anything therein relevant to this limitation. In this regard, as mentioned above, the Office Action incorrectly called the OOTransport class of the OO generic device interface 54 as the "second device driver" required by the claims. As a result, the portions of Dinallo cited by the Office Action actually pertain to the device interface 54, rather than any device driver, and they could not provide the necessary teaching regarding the claim limitation concerning the "second device driver." If the Examiner should maintain the same ground of rejection in the next Office Action, the Examiner is respectfully requested to clearly explain how the cited portions of Dinallo could teach this limitation, so that a clear record is available to facilitate an appeal.

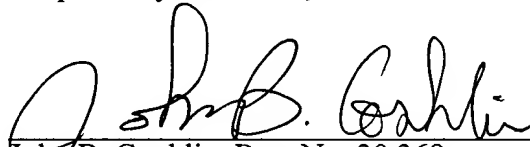
Since the claim elements and limitations are not properly found in the Dinallo reference as discussed above, the Section 103 rejection of independent claims 1, 8, 12 and 16 is not well-supported and should therefore be withdrawn. Accordingly, these independent claims should be allowable. Since the remaining claims all depend from the independent claims, they should also be allowable.

In re Appln. of Sadovsky et al.  
Application No. 09/809,237

*Conclusion*

The application is considered in good and proper form for allowance, and the Examiner is respectfully requested to pass this application to issue. If, in the opinion of the Examiner, a telephone conference would expedite the prosecution of the subject application, the Examiner is invited to call the undersigned attorney.

Respectfully submitted,

A handwritten signature in black ink, appearing to read "John B. Conklin", is written over a horizontal line.

John B. Conklin, Reg. No. 30,369  
LEYDIG, VOIT & MAYER, LTD.  
Two Prudential Plaza, Suite 4900  
180 North Stetson Avenue  
Chicago, Illinois 60601-6780  
(312) 616-5600 (telephone)  
(312) 616-5700 (facsimile)

Date: December 22, 2004